UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/700,224	11/03/2003	John H. Sandham	1801270.00131US1	3106

23483 7590 05/29/2008

WILMERHALE/BOSTON
60 STATE STREET
BOSTON, MA 02109

EXAMINER	
VO, TED T	

ART UNIT	PAPER NUMBER
2191	

NOTIFICATION DATE	DELIVERY MODE
05/29/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

michael.mathewson@wilmerhale.com
teresa.carvalho@wilmerhale.com
sharon.matthews@wilmerhale.com

Office Action Summary	Application No.	Applicant(s)	
	10/700,224	SANDHAM ET AL.	
	Examiner	Art Unit	
	TED T. VO	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 February 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *See Continuation Sheet* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 3-7, 10-11, 13-20, 22-34, 36-40, 43-44, 46-53, 55-67, 69-73, 76-77, 79-86, 88-99* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Disposition of Claims: Claims pending in the application are 1,3-7,10,11,13-20,22-34,36-40,43,44,46-53,55-67,69-73,76,77,79-86 and 88-99.

DETAILED ACTION

1. This action is in response to the amendment filed as RCE on 02/11/2008.

Claims 1, 3-7, 10-11, 13-20, 22-34, 36-40, 43-44, 46-53, 55-67, 69-73, 76-77, 79-86, 88-99 are pending in the application.

Response to Arguments

2. In view of the amendment and arguments given in Remarks, filed on 02/11/08, The amendment necessitated new ground(s) of rejections presenting in the Action. The arguments have been considered. See the rationales addressed in the rejections below.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1, 3-7, 10-11, 13-20, 22-34, 36-40, 43-44, 46-53, 55-67, 69-73, 76-77, 79-86, 88-99 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

- Per Claims 1, 3-7, 10-11, 13-19:

Claim 1 recites "dividing subject code into a plurality of blocks" right at the beginning of a step without a suggestion for what it is "subject code". This "subject code" is vague because lacking antecedent basis. Thus, this claim is in indefinite.

Claim 3 is dependent on a claim which is canceled. It implies that claims 3-7 are indefinite.

Claims 3-7, 10-11, 13-19 are indefinite as reasoned the same in the rejection of claim 1 above.

- Per Claims 20, 22-34, 36-40, 43-44, 46-53, 55-66:

The claims are rejected in the same reason for reciting "dividing subject code" which is lacking antecedent basis.

- Per Claims 67, 69-73, 76-77, 79-85: Same rationale as addressed in the rejection of Claims 1, 3-7, 10-11, 13-19 above.

- Per Claims 86, 88-99: The claims are rejected in the same reason for reciting "dividing subject code" which is lacking antecedent basis.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1, 3-7, 10-11, 13-20, 22-34, 36-40, 43-44, 46-53, 55-67, 69-73, 76-77, 79-86, 88-99 are rejected under 35 U.S.C. 102(b) as being anticipated by Barrio, "Study of the techniques for emulation programming", The Technical University of Catalonia (UPC), Europe, pp. 1-152, 6-2001 (http://personals.ac.upc.edu/vmoya/docs/emuprog.pdf).

As per Claim 1: Barrio discloses,

A method of verifying program code conversion performed by an emulator, comprising the step of:

a) dividing subject code into a plurality of blocks and executing one of the blocks of subject code through an emulator in a process image on a subject processor according to an emulation context up until a comparable point in the subject code to provide an emulated machine state; (See p. 18-19. both sec. 2.1 and sec. 2.2: see, See the program in p. 18, an Interpreter emulator, that uses case switch to divide the different opcodes ('subject code'). These verifying program code of opcodes (emulator) are divided by the Interpreter emulator, and each OPCODEi() is executed till the point break (*'a comparable point'*). Further see "translation is performed in blocks of code" in p. 19)

b) performing a context switch to a native context (e.g. OPCODEi()) ***and executing the same block of subject code natively in the same process image on the same subject processor up until the same comparable point in the subject code to provide a native machine state*** (See the program call OPCODEi(), that call to a native state of the opcode. See sec, 2.1, and 2.2); ***and***

c) comparing the native machine state from execution of the one block of subject code natively on the subject processor against the emulated machine state from execution of the same block of subject code on the same subject processor through the emulator at the comparable point in the subject code;
wherein the native machine state includes a native memory image and the emulated machine state includes an emulated memory image and the step (a) and/or the step (b) includes selectively isolating access to a memory associated with the subject processor to obtain the native memory image and/or the emulated memory image, respectively.
See page 18, including the program, and see the beneath paragraph of the program. Also see p. 24, sec. 8: Testing the emulator.

As per Claim 20: The rejection is applied the same as the rejection of Claim 1.

As per Claim 26: The rejection is applied the same as the rejection of Claim 1.

As per Claim 30: The rejection is applied the same as the rejection of Claim 1.

As per Claim 3: This claim is indefinite. Barrio discloses, *The method of claim 2, comprising performing the step (a) prior to performing the step (b).* (Incorporated with the rejection of claim 1, Barrio's program shows it OPCODEi() execution is prior to a context switch).

As per Claim 4: Barrio discloses, *The method of claim 3, wherein: the step (a) further comprises providing an emulated image of the subject processor* (Still referred to the program of p. 18, and sec. 2.1, and 2.2, and further see testing the emulator in p. 24, sec. 8)*; the step (b) further comprises providing a native image of the subject processor following the native execution of the program code* (i.e. real result run on real CPU emulator (p. 24, or p. 62)*, following the native execution of the program code; and the step (c) further comprises comparing the emulated image of the subject processor against the native image of the subject processor* (See p. 24, and p. 62, i.e. 'compared')

As per Claim 5: Barrio discloses, *The method of claim 4, wherein the step (a) comprises providing the emulated image of the memory in a load/store buffer associated with the memory, such that the memory is not affected by executing the subject code through the emulator* (i.e. the OPCODEi() defined by the program according to the CPU emulation: *emulated image of the memory*, and see p. 22 first paragraph).

As per Claim 6: Barrio discloses, *The method of claim 5, wherein the emulated image of the subject processor includes an image of one or more registers* (i.e. CPU emulation, p. 22).

As per Claim 7: Barrio discloses, *The method of claim 6, wherein the emulated image of the subject processor includes an image of one or more condition code flags* (e.g. figure 23, p. 48).

As per Claim 10: Barrio discloses, *The method of claim 1, comprising selectively switching between the emulation context* (See Case Switch program) *for running the emulator on the subject processor, a target execution context for executing target code produced by the emulator on the subject processor, and the native context where the subject code runs natively in the subject processor* (e.g.. p. 18, sec. 2.1, and p. 43, paragraph 'The getContext()…'))

As per Claim 11: Barrio discloses, *The method of claim 10, wherein both the native context and the emulation context employ a single image of the subject code* (i.e., testing an emulator. Note: when testing an emulator by running against a sample input code, versus a trusted emulator, the running code for two emulators must be the same, i.e., *employ a single image of the subject code*).

As per Claim 13: Barrio discloses, *The method of claim 1, comprising selecting between two or more verification modes, and dividing the subject code into the plurality of blocks according to the selected verification mode* (refer to "basic block' in the reference, e.g., sec. 2.2).

As per Claim 14: Barrio discloses, *The method of claim 13, comprising repeating the executing and comparing steps for each of the plurality of blocks* (refer to the program of p. 18).

As per Claim 15: Barrio discloses, *The method of claim 14, wherein in a first verification mode each block comprises a single instruction of subject code; in a second verification mode each block comprises a basic block comprising a sequence of instructions from a unique entry instruction to a unique exit instruction; and in a third verification mode each block comprises a group block comprising a plurality of the basic blocks* (refer to program in p, 18, and see sections 2.1, 2.2, and 2.4).

As per Claim 16: Barrio discloses, *The method of claim 1, comprising the steps of: dividing a large segment of the subject code into a plurality of smaller blocks, each block containing one or more instructions from the large segment of subject code; and performing a verification comparison at a block boundary between each pair of consecutive neighbouring blocks in the plurality of blocks* (refer to program in p, 18, and see sections 2.1, 2.2, and 2.4).

As per Claim 17: Barrio discloses, *The method of claim 16, comprising the steps of: providing the subject processor in the emulation context, where control of the processor rests with the emulator, performing program code conversion on a current block BBn to produce a corresponding block of converted target code, and patching an immediately preceding block of subject code BBn-1 with a return jump; executing a context switch routine to enter the native*

context, and executing the immediately preceding block of subject code BBn-1 natively by the subject processor, such that the executing step terminates with the return jump; executing a context switch routine to return to the emulation context, and performing the verification comparison by comparing a native machine state representing the subject processor following execution of the immediately preceding block BBn-1 with an emulated machine state representing a virtual model of the subject processor held by the emulator following execution of the immediately preceding block BBn-1; executing a context switch to a target execution context, and modelling execution of the target code corresponding to the current block of subject code BBn in the virtual model of the subject processor held by the emulator, thereby leaving the virtual model in a machine state representing the end of the current block BBn; and repeating the above steps for each subsequent block in the plurality of blocks, unless the verification comparison reveals an error in the program code conversion.

See p. 24-25, "Testing the emulator", where the emulator is a CPU emulator (p. 26) and including CPU emulator core (p. 42. 43, and 44)).

As per Claim 18: Barrio discloses, *The method of claim 17, further comprising restoring the immediately preceding block BBn-1 to remove the return jump* (only control flow of instruction operations, in a jump table).

As per Claim 19: Barrio discloses, *The method of claim 1, further comprising the steps of: selecting a block of the subject code; executing the block of subject code on the subject processor through the emulator; and appending a return jump to the block of subject code, and executing the block of subject code natively on the subject processor terminating with the return jump, such that the return jump returns control of the processor to the emulator* (e.g., see Fig. 13, p. 34).

As per Claim 22: Barrio discloses, *The method of claim 20, comprising selectively inhibiting access to the memory when executing the target code, such that an emulated memory image is provided in the load/store buffer.* See rationale addressed in the rejection of Claim 5.

As per Claim 23: Barrio discloses, *The method of claim 20, wherein after the program code conversion performed by the emulator running on the subject processor has been verified, the method further comprising the step of: comparing execution of the subject code through the emulator running on the subject processor against execution of the subject code through a second emulator running on a target processor* (See sec. 2.1, 2.2, in pages 18-19, and see p. 24, and p. 62, comparing the execution of the tested CUP emulator to the trusted CPU emulator).

As per Claim 24: Barrio discloses, *The method of claim 23, comprising providing a first host processor as the subject processor, and providing a second host processor as the target processor* (tested CPU emulator and trusted CPU emulator).

As per Claim 25: Barrio discloses, *The method of claim 24, wherein the subject code is natively executable on the subject processor whilst not being natively executable on the target processor* (See rationale addressed in the rejection of claim 3).

As per Claim 27: Incorporated to the rejection of claim 26, Barrio discloses, *The method of claim 26, comprising the steps of: performing a first program code conversion of the subject code including providing a first virtual model of the subject processor in the first emulator, and comparing the first virtual model against the subject processor; and performing a second program code conversion of the subject code including providing a second virtual model of the*

subject processor in the second emulator, and comparing the first virtual model in the first emulator against the second virtual model in the second emulator (See p. 10, sec. 2, CPU emulator is a virtual machine).

As per Claim 28: Barrio discloses, *comprising providing a single way communication from the first emulator to the second emulator.* (See testing the emulator, p. 24, particularly, the emulator is CPU emulator such as VM (p.10)).

As per Claim 29: Barrio discloses, *The method of claim 27, comprising the steps of: synchronizing the first and second virtual models by sending initial state information from the first emulator to the second emulator* (i.e. after getting the status from the CPU emulator, its result will be compared the real result)*; for each block of subject code, executing the block of subject code through the first emulator and providing a set of subject machine state data and non-deterministic values to the second emulator; executing the block of subject code in the second emulator substituting the non-deterministic values and providing a set of target machine state data; and comparing the subject machine state data against the target machine state data and reporting an error if a divergence is detected, otherwise repeating the process for a next block of subject code* (See p. 18-20, and sec 2.2: 'Basic block in compiler theory').

As per Claim 31: Incorporated to the rejection of Claim 30, Barrio discloses, *The method of claim 30, wherein the dividing step comprises dividing the subject code the subject code such that each of the plurality of blocks of subject code contains a single instruction* (See program in

p. 18, see p. 24, sec. 8, second paragraph, "instructions", or testing some of the parts of the emulation, last paragraph).

As per Claim 32: Barrio discloses, *The method of claim 31, wherein after program code conversion performed by the second emulator is verified for each of the plurality of blocks of subject code containing a single instruction, the method further comprises: secondly repeating the step (a) by redividing the subject code into a plurality of new blocks, wherein each new block is a basic block comprising a sequence of instructions from a unique entry instruction to a unique exit instruction; and repeating steps (b)-(e) for each basic block, thereby verifying program code conversion performed by the second emulator for every basic block of subject code* (See program in p. 18, and sec. 2.1, sec. 2.2, and see p. 24).

As per Claim 33: Barrio discloses, *The method of claim 32, wherein after program code conversion performed by the second emulator is verified for each of the basic blocks of subject code, the method further comprises: thirdly repeating the steps (a) by redividing the subject code into a plurality of group blocks, wherein each group block comprises a plurality of basic blocks; and repeating the steps (b)-(e) for each group block, thereby verifying program code conversion performed by the second emulator for every group block of subject code.* (See the program of p. 18, and sec. 2.1, 2.2).

As per Claims 34, 36-40, 43-44, 46-52: Claims recite a computer-readable storage medium that stores instruction performing the steps of claims 1, 3-7, 10-11, 13-19. See rejection addressed in Claims 1, 3-7, 10-11, 13-19.

As per Claims 53, 55-58: Claims 53, 55-58 recite a computer-readable storage medium that stores instruction performing the steps of claims 20, 22-25. See rejection addressed in Claims 20, 22-25.

As per Claims 59-62: Claims 59-62 claim a computer-readable storage medium that stores instruction performing the steps of claims 26-29. See rejection addressed in Claims 26-29.

As per Claims 63-66: Claims 63-66 claim a computer readable storage medium that stores instruction performing the steps of claims 30-33. See rejection addressed in Claims 30-33.

As per Claims 67, 69-73, 76-77, 79-85: Claims recite an emulator apparatus that performs the steps of claims 1, 3-7, 10-11, 13-19. See rejection addressed in Claims 1, 3-7, 10-11, 13-19.

As per Claims 86, 88-91: Claims recite an emulator apparatus that performs the steps of claims 20, 22-25. See rejection addressed in Claims 20, 22-25.

As per Claims 92-95: Claims 92-95 claim an emulator apparatus that performs the steps of claims 26-29. See rejection addressed in Claims 26-29.

As per Claims 96-99: Claims 96-99 claim an emulator apparatus that performs the steps of claims 30-33. See rejection addressed in Claims 30-33.

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The

examiner can normally be reached on 8:00AM to 4:30PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is assigned is the Central Facsimile number **571-273-8300**.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TTV
May 23, 2008

/Ted T. Vo/
Primary Examiner, Art Unit 2191